My scriptwriting process is iterative. I do it all in my ${EDITOR} in DokuWiki format, then copy the rendered HTML into libreoffice writer to share and print. I don’t normally provide copies of my scripts but I’m providing this here for the audience’s edification. Perhaps this can give some insight into my talk-writing process. I practiced this a couple dozen times at various stages and revisions to iron out all the kinks and get my timings down.

Hello everyone, welcome to SeaGL 2025. My name is Romeo, and this is What Is Free May Never Die. Today, I am going to

unlock an idea that's already in your brain, it's just dormant. By the end of these next 20 minutes, you will have a new way to talk and think about free and open source software. It won't just be

A way to preserve the four essential freedoms or

A way to build better software at a lower cost through crowdsourcing, but as a way

To make your work immortal *PAUSE FOR EFFECT* Let's set out a roadmap of what we're going to cover in our short time together.

We're going to start with a definition of terms and I'm going to tell you exactly how free software lives forever

Then we're going to look at some historical evidence to support this idea.

And to finish up we will cast our gaze into the distant future and I'll show you the one and only thing that we have to do to make sure that someday all software is free.

Starting off with software immortality. Now, immortality is a heavy word. It conjures up

all sorts of different mental images. That's not the kind of immortality that I'm talking about, I'm not Him. I can't offer you that. In fact our modern science-based worldview says that it's impossible

The guys with labcoats and telescopes

look at the sun, and they say that

the sun is going to explode.

They look at the universe and say that it's all going to fizzle out into heat death

They say that nothing is immortal, everything dies.

They say that nothing beats entropy.

I say that's lame. I'm not talking about that kind of immortality. I'm talking about immortality on a human timescale.

This is a handaxe. It's a rock shaped like an almond that is sharp all the way around. It was a general purpose multitool. You could use it to cut, chop, slice, carve, dig, maybe even hunt. It was invented by…actually we don't know who invented it, that's been lost to history.

Fossil evidence says that it probably started in what is now Etheopia

and then spread out to the rest of the planet from there. It was the single most useful and ubiquitous tool on the planet Earth for all of caveman-kind for a short reign of

1 and a half million years *PAUSE FOR EFFECT* Sharp rock. 1 and a half million years. The handaxe was immortal.

This is Isaac Newton. You may have heard of some of his all time classic hits like

Gravity and

The Laws Of Motion or one of his lesser-known works

Discovering That Light Goes Into Eyes Instead Of Coming Out Of Them. Well 361 years ago he was messing around with a pen and paper and invented this little thing called

calculus. Which is a super useful piece of math if you want to have

a modern world. Everything around us requires that humans know calculus. I don't know if humans will exist in another 1.5 million years, but if they do

they won't know who this guy was. He'll have been lost to history.

But they'll know calculus. Calculus is immortal. We can tell that this is true just by seeing its impact in the

361 years since its invention.

This rather dashing fellow is Tom Kilburn. Just a short while ago

On June 21st 1948, he wrote and ran the first ever piece of software that was stored in memory and executed on a processor.

On the Manchester Baby, that computer right there.

We ran the first software on a computer 77 years ago. That's a crazy short amount of time.

I think everyone here knows who this guy is.

We started caring about free software 42 years ago. The blink of an eye in historical terms.

We don't have 1.5 million years of hindsight.

We don't have 361 years of hindsight.

We have 77 years of hindsight.

Or really we have 42 years to look at software's longevity. It's difficult to make grand sweeping statements about what is and isn't possible with such a short amount of time. We have less than a lifetime to look back on.

But software is a tool.

Software is math. And just like this tool and this math, software can be just as immortal if can just avoid death. Death is a funny subject. We like to wax poetic about it.

One of those poetic things that we say is that a man dies twice. The first time when his heart stops beating

and a second time when people stop repeating his name. I like to wax poetic too, and I say that software also dies twice.

Once when it stops being worked on.

And a second time when it stops being used. *PAUSE FOR EFFECT* For most proprietary software projects

Especially modern SaaS offerings, these two deaths happen at roughly the same time. But when the software is free, when the code is there for anyone to continue to use and modify, when

the whole human community has the ability to keep carrying the torch when you stop, these two deaths never come. Death becomes impossible.

Sometimes it seems like impossible things are happening. That bright spot on the right side of the image is actually the sun. The seemingly impossible, sunshine in Seattle. Free software can do something similar to death, but it's not quite death. It's more

of a dormancy. Free software enters hibernation when nobody cares. When it's been modified for the last time and when the last user stops running it, free software enters a slumber. Waiting for someone, anyone

to come pick up the torch and keep it going. For proprietary software, there is a very short list of people who even could pick up the torch, and they're typically legally prohibited from doing so. With free software, anyone can, but someone has to want it.

These are my grandpa's walking shoes.

He's had them for 15 years now, over a third as long as we've had free software at all. He's walked all over the world with these bad boys, and he's maintained them.

He's relaced them twice, resoled them, and replaced the heels. They keep going because someone wants them to keep going. Software is the same.

Alright, we've talked about what software immortality is and how it's attained. Now I'll show you some examples from software's short history where proprietary software has succumbed to the reaper and free software has kept on ticking. **EXAMPLE #1, video game emulators VS the big bad wolf**.

On the proprietary side of the house, Connectix Virtual Game System was a PlayStation emulator that let you play PlayStation games without a PlayStation.

Sony is a big bad wolf and they didn't like that so they threw lawyers at the problem. Connectix actually won in court, but the legal battle was so long and expensive that they collapsed as a company. Sony bought out what was left

and decommissioned the software. No more Connectix. It's dead.

On the libre side of the house, Yuzu was Switch emaultor that let you play Switch games without a Switch.

Nintendo is a big bad wolf and they didn't like that so they threw lawyers at the problem. Maybe you've heard this one beofre. So had Tropic Haze (the company behind Yuzu). They settled out of court, and as part of the settlement

stopped distributing Yuzu and promised never to do it again. No more Yuzu. But Yuzu was free software.

So it came back, and with more than before. Based on Yuzu's source code, there's SuYu, Eden, and Torzu. Because free software belongs to all of humanity, not a company. You can't kill free software by killing a company. **EXAMPLE #2, social network features**.

On the proprietary side of the house, Twitter was a social network where you didn't post, you tweeted. Because you're a bird. It was whimsical and fun, the way the Internet should be.

Someone bought Twitter and changed the Tweets to Posts and now you can't Tweet anymore. Don't like it? Post about it. Because proprietary software features can be taken away.

On the libre side of the house, Mastodon is a social network software where you didn't post, you tooted. Because you're a wooly mammoth. It was whimsical and fun, the way the Internet should be.

And SOMEONE SNITCHED and told the developer that that's a fart joke and now you can't Toot anymore. Don't like it? Post about it. Or just change it back.

It's one commit. Three files changed. You can't take features away in free software,

we can just put them back. You can keep tooting until the wooly mammoths come home. (they're extinct they're not coming home you can toot forever). **EXAMPLE #3, death from old age**

There's this movie from 1993 called Jurassic Park. In it this girl sits down at a computer and says “it's a unix system, I know this”. She's sitting at a Silicon Graphics IRIX machine using

FSN, the File System Navigator. Cool looking file browser, right? I'd like to give it a whirl. I can't. It's gone. I can find a download for it but I can't run it. It only runs on IRIX systems, they stopped selling these in 2006. I don't have one. And I don't have the source code to make it work myself. FSN is dead by being too obscure and by changes in computer architecture. It was proprietary, and it's gone.

In comparison, we have a piece of software called FSV, or File System Visualizer. It was made to be similar to FSN, and you can see the resemblance. You can still download and compile this and use it on the computer in your lap right now. You have to fiddle around with some legacy GTK stuff but you can get it to work. And you will still be able to in another 20 years. Because free software doesn't die by becoming irrelevant. It just goes dormant. **EXAMPLE #4, government censorship**

Red Dot and ICEBlock were two proprietary software apps that allowed people to mark when and where they saw ICE agents so that the people that were being hunted by ICE for could skedaddle.

The US DOJ didn't take too kindly to tech bros disrupting the deportation space, so they did what governments do and

pressured the mobile computing duopoly to take these apps down. These two mega-powerful companies, of course, complied, since you don't get to the top by making enemies in high places.

Bye bye Red Dot and ICEBlock. Centralized service, centralized distribution, killed by government censorship.

I can think of a couple of pieces of free software that the US government and powerful companies would like to censor out of existence. But they can't. The last time they tried,

we put it on a T-shirt. Because you can't use government pressure on big companies to make the population forget free software. **NEXT UP, software sunsetting**

These two will be around forever because they are open protocols and anyone can run their own service. Changes to people's expectations, changes in common practice, clients getting worse for some reason, and the network effect on a global scale might make that mostly irrelevant, but if you and your friends want to run a server and clients that are frozen in time right now, you can do so, and no one can ever take it away from you. You can't sunset something that I can host myself. You know what you can sunset?

PAUSE FOR EFFECT The guys on the right are dead. I will never get the Skype ringtone out of my head. But it's gone. The ones on the left will never go away. What happens if you try? If you make an effort to close down free software? Take a look at Tiny Tiny RSS.

PAUSE TO LET THEM READ Fox, the lead developer, maintained it for 20 years, but he *just* threw in the towel. Last week.

Fox no longer finds it fun. And that's okay.

Greg still finds it a blast. Look at how much fun Greg is having. He's a squirrel with a cup of coffee. Thanks Greg. Now, Fox gave a couple months head's up. But what if he didn't?

This fella went by the screen name, Why The Lucky Stiff.

He wrote this really useful Ruby library named Shoes, along with a bunch of other useful stuff. Then one day out of the blue

He deletes his whole entire web presence and all of the sites and repos. Gone.

Except it was free software, so someone else had a copy of the repo. So Shoes, and Why's other useful projects, still exist. Since 2009 they've been maintained by the community. Because you can't even sunset free software by surprise as long as one person cares. **NEXT EXAMPLE, video games**

This is Doom running on a smart thermostat. Doom was written for x86 processors running MS-DOS. This is a thermostat with an ARM9 processor. This is possible in part due to people's dedication, but in part

because Doom is open source. You can just compile it for anything. So you can put Doom

on a lawnmower. In 200 years our descendants will be running Doom on whatever crazy new architecture they have.

Not Pajama Sam. **LAST EXAMPLE, version control software**

BitKeeper was a version control management software that was invented specifically for use with the Linux kernel. It was proprietary, but provided at no cost to kernel devs on the one condition that they don't try to reverse engineer it and make a competing project. Spoiler alert, they reverse engineered it and made a competing project

it's named git. Once BitKeeper pulled its gratis license from the kernel devs, it was completely usurped by git, and it fell into history's trash bin with all the other proprietary software projects that don't exist anymore.

Actually BitKeeper is technically still around becasue it was open-sourced as a final act. Maybe the author finally saw the clear pattern

Free software keeps living. Proprietary software dies. When proprietary software is on its deathbed and doesn't want to die, it becomes free, and then keeps living. Because what is free may never die. This is obvious from just 42 years of writing free software.

That's enough of the past, let's take a look forward in time and see where we'll wind up with free software immortality.

Here on the Y Axis we have the amount of software, and on the X axis we have time.

The amount of free software can only increase with time

While the amount of nonfree software can increase or decrease. So we just need to keep going and we'll eventually make up the majority of software that there is. That's a little bit disingenuous, there's actually two ways that this graph can look.

This is what our future will look like if the amount of nonfree software, even with its ability to die, outpaces the amount of free software. In order to avoid this fate, we need to maintain a lead on growth. However small. We just need to maintain it forever. I don't know if you've noticed this, but we're really good at maintaining things forever. It's kind of our special skill. If we can keep our lead

then eventually, after hundreds or thousands of years, the proportion of nonfree software will asymptotically approach zero. All of the software will be free. You'll be free, hackers. But only if you keep it up.

So pick up your keyboards. Open your editors.

Go invent calculus. Make a handaxe. And since software is so much more complicated, you'll (inevitably) leave a little bit of yourself in your product.

The fingerprints of your brain will be all over my computer, and through your software's immortality, a little part of you will keep on ticking, until the mastodons come home.

Happy hacking. *Hold for applause or some kind of reaction*. You've all been a great audience way better than my audience last week,

who kept falling asleep during my practice runs. His name is Max. He gets a pass. He was all tuckered out after a long day of being a cat. Really though, you guys have been amazing.

I have been your speaker. That's me. punkto.org is my personal website where you can find my contact info, notes on this talk, these slides, and soon a recording. My name is Romeo.

I gave this talk today because I've never heard anyone else express this thought, but I think that it's worth discussing with other people. I want **your perspectives** on this. I think that it could be a useful framework to encourage people to write and maintain, and maybe even use, free software. I'd love to talk with you about this throughout the conference. I want to let you know in advance

some people see me furrow my brow and they think that I'm mad at them. I'm not mad. That's just a thing that my face does.

If you want to continue this discussion online, I've made a Matrix chatroom for it, which you should be able to join by scanning that QR code. Or you can type in **#immortal:punkto.org**

I think we have time for N questions. / We don’t have time for questions.